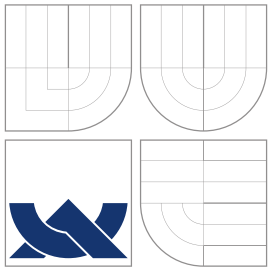
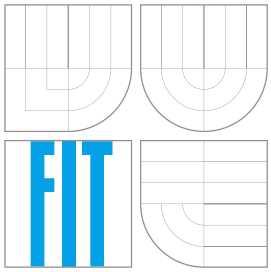


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

HYBRIDNÍ ROZPOZNÁVÁNÍ 3D OBLIČEJE

HYBRID 3D FACE RECOGNITION

ROZŠÍŘENÝ ABSTRAKT DISERTAČNÍ PRÁCE

EXTENDED ABSTRACT OF PHD THESIS

AUTOR PRÁCE — ŠTĚPÁN MRÁČEK

AUTHOR

VEDOUCÍ PRÁCE — MARTIN DRAHANSKÝ

SUPERVISOR

BRNO 2015

Abstrakt

Tato disertační práce se zabývá biometrickým rozpoznáváním 3D obličejů. V úvodu práce jsou prezentovány současné metody a techniky pro rozpoznávání. Následně je navržen nový algoritmus, který využívá tzv. multialgoritmickou biometrickou fúzi. Vstupní snímek 3D obličeje je paralelně zpracování dílčími rozpoznávacími podalgoritmy a celkové rozhodnutí o identitě nebo verifikaci identity uživatele je výsledkem sloučení výstupu těchto podalgoritmů. Rozpoznávací algoritmus byl testován na veřejně přístupné databázi 3D obličejů FRGC v 2.0 i vlastních databázich, které byly pořízeny pomocí senzorů Microsoft Kinect a SoftKinetic DS325.

Abstract

This Ph.D. thesis deals with the biometric recognition of 3D faces. Contemporary recognition methods and techniques are presented first. After that, the new recognition algorithm is proposed. It is based on the multialgorithmic fusion. The input 3D face scan is processed by the individual recognition units and the final decision about the subject identity is the result of combination of involved recognition unit outputs. Proposed approach has been tested on the publicly available FRGC v 2.0 database as well as on our own databases acquired with the Microsoft Kinect and SoftKinetic DS325 sensors.

Klíčová slova

biometrie, 3D obličej, multi-algoritmická fúze.

Keywords

biometrics, 3D face, multi-algorithmic fusion.

Citace

Štěpán Mráček: Hybrid 3D Face Recognition, rozšířený abstrakt disertační práce, Brno, FIT VUT v Brně, 2015

Contents

Chapter 1

Introduction

Face recognition is one of the most frequently used biometric techniques. In everyday life, we recognize other people by their faces. We are able to localize a face in a very large and complicated scene. Also the detection of anatomical features, like nose, eyes, and mouth position within the face, does not pose us difficulties. Furthermore, we can recognize faces from various angles, even if face expressions are present or a part of a face is covered. Many activities that we do completely automatically with no effort become quite difficult if we try to describe this process mathematically.

Nevertheless, a lot of research has been done in the area of the biometric face recognition, especially in the three-dimensional recognition in recent years. The 2D face biometric has become together with fingerprints a part of biometric passports in the European Union and all member states of the ICAO (*International Civil Aviation Organization*). Another biometric modality that is used in the biometric passports is the iris. The face was recommended as the primary biometrics, mandatory for global interoperability in passport inspection systems, while the finger and iris were recommended as secondary biometrics to be used at the discretion of the passport-issuing state [18].

The biometric face recognition, which is the main focus of this work, has a wide application in practice, e.g. the biometric passports, as was mentioned above, or in access control systems. Because of its nature, which is very similar to the way we usually recognize each other, it is very well accepted by users. No special activity is required by the data subject and the recognition process is non-intrusive, which means that the data subject is not in the direct contact with the sensor.

This work is the extended abstract of my Ph.D. thesis that deals with the biometric face recognition and all connected matters. In the second chapter, a proposal of the 3D face recognition algorithm, which is the main goal of this work, is described. The third chapter contains an evaluation of proposed algorithm and describes the achieved results.

Chapter 2

Proposal of the Recognition Algorithm

This chapter describes the generalized recognition pipeline that takes the input face mesh and normalizes it, such that the rotation is compensated for. After that, some image representations of the surface, texture, and curvature are generated from the normalized mesh. The pipeline continues with the application of specific image filters. Finally, subspace projections are used in order to extract features.

The main idea of the proposed method is the score-level fusion of involved individual recognition units. By the application of some filter bank, e.g. Gabor filter bank, on the input image we obtain m new images. However, the number of filters within the bank is quite high, therefore some optimization selection method is needed in order to improve speed as well as remove redundancy. We employ hill-climbing selection and the optimization criterion is fusion EER.

The similar approach has been proposed by Yang et al. [44]. Yang et al. use AdaBoost to select a small set of Gabor features (weak classifiers) in order to form a strong classifier. Moreover, he proposed intra-face and extra-face difference space to transform a multi-class classification to a binary decision. The task is to assign the input two images to intra-personal or extra-personal space.

Su et al. [37] came with an algorithm exploiting both local and global features. The global features are extracted from the whole face images by keeping the low-frequency coefficients of Fourier transform, which he believe encodes the holistic facial information, such as the facial contour. For local feature extraction, Gabor wavelets are applied on the face image patches. The resulting classifier is based on the hierarchical feature-level fusion utilizing *Linear Discriminant Analysis*. However, both methods from Yang as well as from Su are designated for 2D face recognition. Our proposed method is able to profit from both 2D texture and 3D shape data.

2.1 Generalized Recognition Pipeline for Face Recognition

Biometric recognition pipeline usually consists of data acquisition, preprocessing, feature extraction, and comparison that yields to the final decision whether the user is accepted or not [43].

The recognition pipeline suitable for the face recognition presented in this thesis is depicted in Figure 2.1. Individual components are described in the following sections.

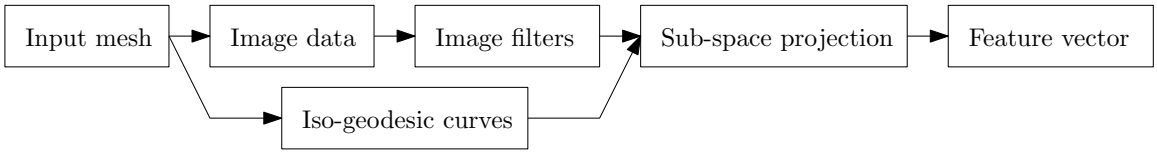


Figure 2.1: Recognition pipeline for 3D face recognition.

First, the input mesh is aligned. One can use an automatic landmark detection followed by translation and rotation of the detected points to some predefined position. Another approach might be the ICP align of the entire input face mesh to a reference template.

Image data are extracted from the aligned face mesh. The mesh is transformed to the range image and texture representations. The range image is further processed in order to gain 4 new curvature representations – mean curvature, Gaussian curvature, eigencurvature and shape index image.

The alternative to image filters approach are the iso-geodesic curves. A set of curves centering at a given point is retrieved from the mesh and converted to a set of points.

The next step, common to both iso-geodesic curves as well as image filters, is some subspace projection of the input data. Image matrix is transformed to the column vector representation and projected to the low-dimensional space after that. A set of 3D points is transformed to the simple column vector in the same manner. PCA [41], ICA [45], as well as LDA [3] are suitable for a subsequent subspace projection of the input column vector.

2.2 Face Alignment

The proper face alignment is a crucial part of the input face pre-processing. Here, we use an alignment based on the reference face template. Input face mesh is aligned to the template such that the sum of square differences between the input face mesh and corresponding points on the template is minimal. In the following subsections, the creation of the reference face template and the alignment itself will be described.

2.2.1 Reference Template Creation

The reference face template was created from 100 face scans taken from the FRGC database. Nine points were manually annotated on each scan – 2 outer eye corners, 2 inner eye corners, nasal bridge, nose tip, outer nose corners, and lower nose corner. *Procrustes analysis* [13] was used in order to align annotated points as well as corresponding scans. The algorithm operates in the following steps:

1. Translate each scan so that its center of gravity (CoG) is at the origin.
2. Arbitrarily choose one example as an initial estimate of the mean.
3. Align all scans (using translation and rotation) with the mean.
4. Re-estimate the mean from aligned scans.
5. If not converged, return to 3.

The key point during the Procrustes analysis is the alignment of all scans to the reference mean face template. The translation is quite simple – scan is moved in such direction that its CoG merges with the CoG of the reference mean scan.

Figure 2.2: Mean reference face template used for proper registration of input faces.

The rotation is performed using *Singular Value Decomposition* (SVD). Let the $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_9)$ denotes a set of individual annotated landmarks $\mathbf{x}_i = (x_{xi}, x_{y_i}, x_{zi})^T$. The similar 3×9 matrix Y denotes landmarks from the mean face scan. First, the covariance matrix S is calculated:

$$S = XY^T \tag{2.1}$$

The next step is singular value decomposition of S. It seeks for real matrices U and V and for diagonal matrix Σ, such that it holds the equation:

$$S = U\Sigma V^T \tag{2.2}$$

The optimal rotation matrix R from X to Y is finally computed:

$$R = V \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & \det(VU^T) \end{pmatrix} U^T. \tag{2.3}$$

Individual aligned face scans were converted into a range image representation (see Section 2.3) and the mean range image was computed. The resulting mean range image was cropped such that it contains only the eyes, nose, upper part of mouth, and cheeks. The range image was sub-sampled to the resolution 17×21 pixels and converted to the mesh representation again. The resulting face alignment reference template is shown in Figure 2.2.

2.2.2 Iterative Closest Point Alignment

The main task of the *Iterative Closest Point* (ICP) algorithm is to align (using translation and rotation) a scanned face to the reference face mesh [5]. Although the scans in our testing databases were taken when the subjects were front-facing the capturing device, variations in rotation and even more in position are present.

The algorithm steps are:

1. Associate all points in the reference template with the corresponding points from the input face using the nearest neighboring criteria.
2. Compute an optimal translation and rotation using least-squares as the optimization criteria.
3. Translate and rotate the input face mesh.
4. Iterate until convergence.

The first problem is that, contrary to the Procrustes analysis align, there is no explicit mapping between the points of the input face and the reference template. Moreover, the scans in FRGC database contain up to 80,000 vertices. The exhaustive linear search for the nearest matching neighbor is thus ineffective. *Fast Linear Approximation of Nearest Neighbor* (FLANN) is therefore used [27].

The ICP algorithm may fail if the initial position of the input face requires significant translation. The corresponding points are then wrongly estimated and the convergence is not assured. In order to avoid this, some rough estimation of the initial translation has to be supplied. We use a simple template matching. The input face mesh is gradually moved over the reference template and a sum of square differences is calculated between template points and appropriate points on the input mesh. The rough initial position estimation is at the point where the sum of square differences is minimal.

The k-means index is used for the nearest neighbor search during the computation of the optimal translation and rotation. The problem is that the input face mesh points index has to be re-calculated after each ICP iteration. However, this issue can be solved with a simple trick: In each ICP iteration the inverse translation and rotation is added to the temporary stack. When there is a need to gain a point p on the reference face mesh that is the nearest to some arbitrary point on the input face, the inverse transformation is applied on the point p first.

2.3 Source Image Data

Although many 3D face recognition algorithms operate directly in three-dimensional space (see [13, 19]) we propose an approach that converts the input 3D face mesh into a 2D matrix on which the subsequent recognition (image filters and feature extraction) operates. Since we have texture as well as the 3D model, several representations describing the texture, depth, and curvature may be deduced.

The range image (depthmap) is created from the input face scan in several steps: First, the point-cloud representation is transformed to the triangular mesh using Delaunay triangulation [16]. After that, the mesh vertices are projected to the x-y plane and the z-coordinate is transformed to a pixel brightness. The brightness of the remaining points within the triangles is linearly interpolated. The Pineda algorithm [30] is used for fast triangle rasterization. The resulting range image is slightly smoothed with Gaussian kernel in order to soften the edges between the triangles.

The remaining images of the surface representation depend on the calculation of the principal curvatures. Curvature k at each point B on the range image is calculated from the z-coordinate b_z of the point B as well as from its surrounding points A and C and their z-coordinates a_z and c_z respectively (see Figure 2.3). The curvature is approximated as the signed angle $\alpha = \pi - |\angle ABC|$. Its sign is deduced from the comparison of b_z and $d_z = \frac{a_z + c_z}{2}$. If the $b_z < d_z$ then the sign is negative. The principal curvatures k_1 and k_2 are estimated in x axis as well as in y axis direction and swapped eventually such that $k_1 > k_2$.

Despite the mean curvature, Gaussian curvature, and shape index, *eigencurvature* [36] is used. It is computed directly from the image point $\mathbf{P} = (p_x, p_y, p_z)$ and its 8 surroundings $(\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_8)$. It is based on the PCA of the matrix M:

$$M = \begin{pmatrix} \mathbf{P} & \mathbf{P}_1 & \cdots & \mathbf{P}_8 \end{pmatrix} = \begin{pmatrix} p_x & p_{1_x} & \cdots & p_{8_x} \\ p_y & p_{1_y} & \cdots & p_{8_y} \\ p_z & p_{1_z} & \cdots & p_{8_z} \end{pmatrix} \tag{2.4}$$

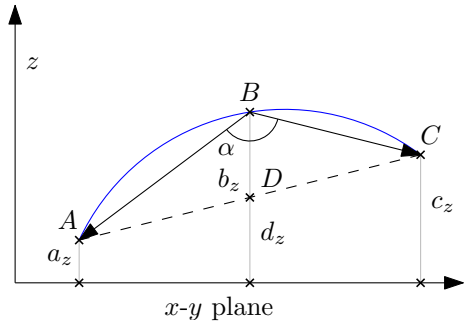


Figure 2.3: Principal curvatures estimation.

Figure 2.4: Image representations of the face surface. From left to right: texture, range image, mean curvature, Gaussian curvature, shape index image, eigencurvature.

The PCA reveals 3 eigenvectors and their corresponding eigenvalues l_0, l_1, and l_2 ($l_0 > l_1 > l_2$). The *eigencurvature* $E_{\mathbf{P}}$ is then:

$$E_{\mathbf{P}} = \frac{l_2}{l_0 + l_1 + l_2} \tag{2.5}$$

The examples of curvature representation images could be found in Figure 2.4.

2.4 Filter Banks

The image filter banks are widely used technique in the area of texture analysis, segmentation, and classification. Individual filters in the bank are used in order to remove unwanted components or features. The two-dimensional filter (kernel k with size $k_w \times k_h$) is convoluted with the input image i and the response d using the following equation is thus calculated:

$$d(x,y) = \sum_{0 \le x' \le k_w} \sum_{0 \le y' \le k_h} = k(x',y') \cdot i(x + x' - a_x, y + y' - a_y) \tag{2.6}$$

The $\mathbf{a} = (a_x, a_y)$ is the kernel anchor – center of the kernel and is usually set to $\mathbf{a} = (k_w/2, k_h/2)$. In fact, the Equation 2.6 does not compute the real convolution since the kernel is not mirrored around the anchor point. The image filter banks are set of m 2D kernels that are convoluted with the input image (see Figure 2.5).

2.4.1 Gabor Filter Bank

The complex Gabor filter is defined as the product of a Gaussian kernel and a complex sinusoid:

$$g(x,y,\omega,\theta,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x'^2+y'^2}{2\sigma^2}} \left(e^{i\omega x'} - e^{-\frac{\omega^2\sigma^2}{2}} \right) \tag{2.7}$$

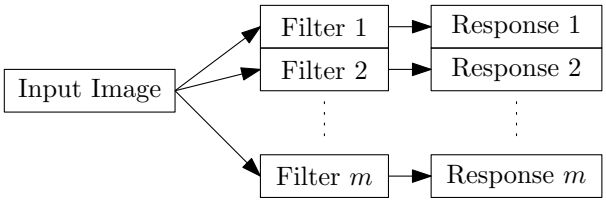


Figure 2.5: General filter bank.

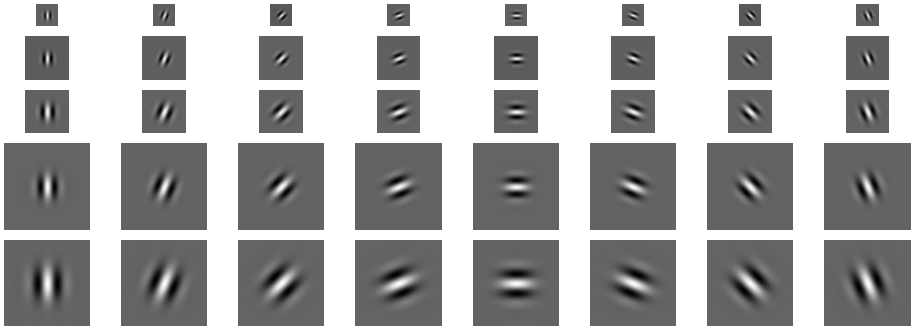


Figure 2.6: Gabor filter bank.

where x and y are coordinates within the Gabor kernel, $x' = x\cos\theta + y\sin\theta$ and $y' = -x\sin\theta + y\cos\theta$. It is often used as an image edge detector with respect to specific frequencies and orientations. The Gabor space is very useful in image processing applications such as optical character recognition [10], iris recognition [7] and fingerprint recognition [11]. The complex sinusoid is known as the *carrier* and the Gaussian-shaped function is known as the *envelope*. The rotation as well as the frequency of the carrier is controlled through the parameters θ and ω, respectively. The parameter σ controls the envelope size.

The Gabor filter is usually controlled with just two discrete-value parameters – orientation $o \in (0, 1, \ldots, 7)$ and scale $s \in (1, 2, \ldots, 7)$. The parameters ω, θ, and σ are set to: $\omega \leftarrow \frac{\pi}{2}\sqrt{2}^{-s}$, $\sigma \leftarrow \frac{\pi}{\omega}$, and $\theta \leftarrow \frac{o\pi}{8}$. The example of Gabor filter bank is in Figure 2.6.

The Figure 2.7 shows the application of the complex Gabor filter on the input shape index image. The Figure 2.8 shows the superposition ability of Gabor filter bank.

(a) (b) (c) (d)

Figure 2.7: Application of the complex Gabor filter on the input shape index representation of the face surface. From left to right: input image (a), real kernel and imaginary kernels (b), real response and imaginary responses (c), and absolute response (magnitude) with angle response (d).

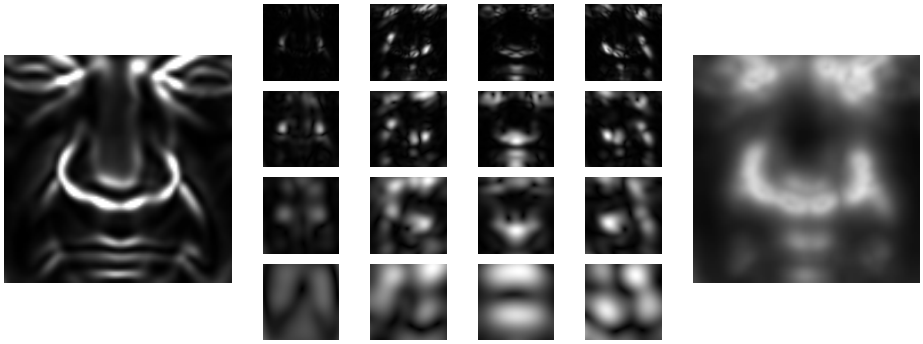


Figure 2.8: Superposition of individual absolute responses of complex Gabor filter. 16 different Gabor filters were applied on the input eigencurvature image (left). The responses are shown in the center grid. Four different frequencies (one for each line) and four different orientations (one for each column) were applied. The resulting superposition is shown on the right side of the figure.

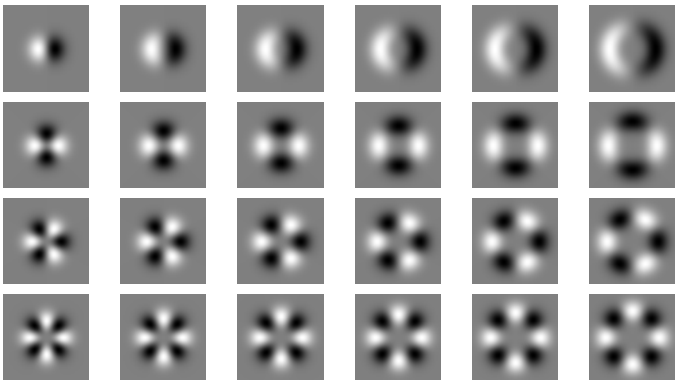


Figure 2.9: Gauss-Laguerre filter bank.

2.4.2 Gauss-Laguerre Filter Bank

The Gauss-Laguerre wavelets are polar-separable functions with harmonic angular shape. They are steerable in any desired direction by simple multiplication with a complex steering factor and as such they are referred to as self-steerable wavelets [1]. Our Gauss-Laguerre filter bank consists of 35 filters that were created with parameters $n \in (1, 2, 3, 4, 5)$, $k = 0$, $j = 0$ with sizes 16×16, 24×24, 32×32, 48×48, 64×64, 72×72, and 96×96 pixels. The θ has been set to $\theta \leftarrow \text{atan2}(x, y)$ and $r \leftarrow \sqrt{x^2 + y^2}$. See Figure 2.9 for an example of Gauss-Laguerre filter bank.

2.4.3 Other Filters

Histogram Equalization

The histogram equalization [35] may improve face recognition based on the classic 2D photographs or on thermal imaging. It improves the contrast in an image in order to stretch out the intensity range. Equalization implies mapping one distribution (the given histogram) to another distribution (a wider and more uniform distribution of intensity values) so the intensity values are spread over the whole range. Figure 2.10 shows the

Figure 2.10: The example of application of histogram equalization filter. The upper row contains photographs of the same subject in different lighting conditions. The lower row contains the same scans with histogram equalization applied.

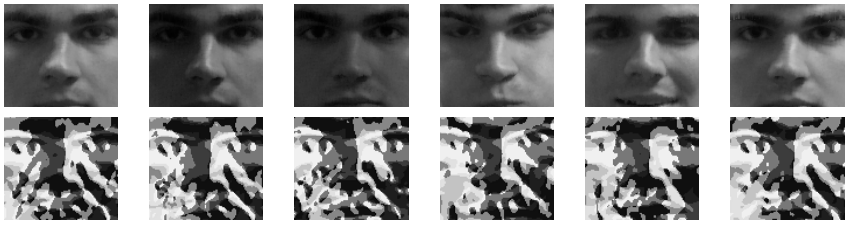


Figure 2.11: The example of application of the LBP filter. The input images are in the first row while the results after filter application are in the second row.

impact of histogram equalization on the set of images that belong to the same subject but the lighting conditions vary.

Gaussian blur and difference of Gaussians

Gaussian blur filter may improve recognition robustness against noise and wrong face alignment. The *Difference of Gaussians* (DoG) [39] works as a bandwidth filter. Shading induced by surface structure is a potentially useful visual cue but it is predominantly low spatial frequency information that is hard to separate from effects caused by illumination gradients. Suppressing the highest spatial frequencies potentially reduces both aliasing and noise without destroying too much of the underlying recognition signal.

Local Binary patterns

The *Local Binary Pattern* (LBP) [40] operator labels the pixels within image by thresholding the 3×3 neighborhood with the center value and considering the result as a binary number. At a given pixel, LBP is defined as an ordered set of binary comparisons of pixel properties between the center pixel and its eight surrounding pixels. The decimal form of the resulting 8-bit word (LBP code) is used to represent the detail property of the center pixel. The example of application of LBP filter is in Figure 2.11.

Local binary patterns are often used with spatial histograms – the image is divided into grid and histograms are calculated within each cell. Concatenated histograms may form the feature vector directly or they can be further processed with some subspace projection.

Figure 2.12: Iso-geodesic curves.

2.5 Iso-geodesic curves

3D Face recognition utilizing iso-geodesic curves and iso-geodesic stripes has been described in [4, 13, 9]. The extracted curves are sampled to a set of points in 3D space and directly processed by some subspace projection technique, i.e., PCA, LDA, and ICA.

The center of all facial iso-geodesic curves is the nose-tip N that has been previously located during the ICP align. n curves $c_1, c_2, \ldots, c_n$ with corresponding geodesic distances $d_1, d_2, \ldots, d_n$ are thus extracted. Each curve c_i consists of m points: $c_i = (p_{i_1}, p_{i_2}, \ldots, p_{i_m})$, such that $d_{geo}(N, p_{i_j}) = d_i$, where $d_{geo}(\cdot, \cdot)$ denotes geodesic distance.

The individual points p_{i_j} of curve c_i are gained in the following manner: The neighborhood of the center is equally divided into m sectors such that the angle between individual sector beams is $\frac{2\pi}{m}$. On each sector beam, a point with a specific geodesic distance from the center is denoted. Example of 5 iso-geodesic curves with geodesic distance 1, 2, 3, 4, and 5 cm consisting of 100 points is shown in Figure 2.12.

2.6 Feature Extraction and Metric Selection

Although specific Gabor filter may reveal features important for the subject classification, the dimensionality of the face image space remains the same. Moreover, if we apply 10 Gabor filters on the image with size 50×50 pixels, the resulting dimensionality is $50 \cdot 50 \cdot 10 = 25\,000$. Therefore, the image is projected to some low-dimensional space using som subspace projection technique.

In plain PCA, the components of the projected vector are proportional to the variability that is expressed as the corresponding eigenvalue. This unbalance of individual feature vector components may lead to a neglect of those feature vector components that may have positive impact on the recognition performance, however, their associated eigenvalue is too small. In order to avoid that, individual feature vector components can be normalized after the subspace projection using z-score normalization. That is, an arbitrary feature vector $X = (x_1, x_2, \ldots, x_m)$ is modified such that $x_i \leftarrow \frac{x_i - \bar{x}_i}{\sigma_i}$, where $\bar{x}_i$ is the mean value of the component i and σ_i is corresponding standard deviation – see Figure 2.13.

Usually, the basic Euclidean distance is used in order to compare two feature vectors. We have tried other metric functions as well – namely a sum of absolute differences (city-

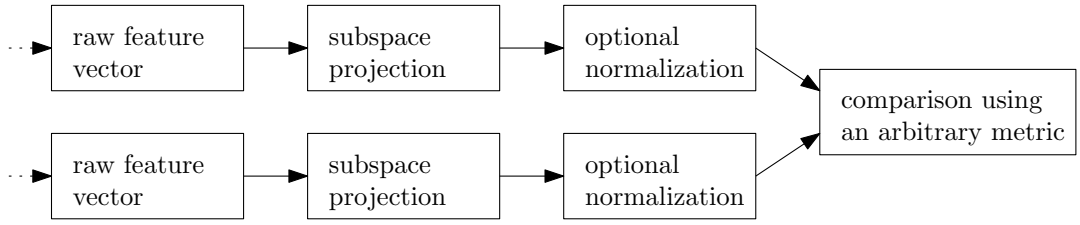


Figure 2.13: End of the recognition pipeline – raw feature vectors (iso-geodesic curves, processed images) are projected to low-dimensional space and optionally normalized using z-score normalization. The comparison is conducted using an arbitrary distance function.

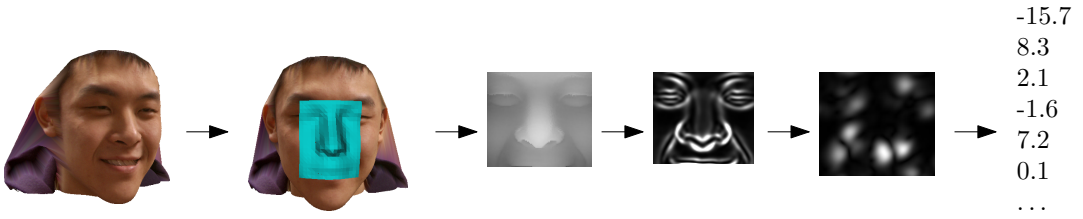


Figure 2.14: The example of one possible recognition unit. The input face mesh is aligned first. After that, the range image and subsequently eigencurvature is calculated. On the curvature image, Gabor filter is applied. The subspace projection using PCA is made as the last step.

block, Manhattan metric), cosine metric, and correlation metric.

The alternative to the PCA projection approach is the utilization of spatial histograms that are closely related to the LBP-based recognition [39]. After the application of the LBP filter, the image is divided into the grid. The grid cell size depends on the specific application as well as on the size of the input image. The image is converted to the grayscale representation and the histogram of the intensity values is calculated. Individual histogram values from all grid cells are concatenated and the resulting feature vector is thus created. The size d of the feature vector depends on the cell count: $d = r \cdot c \cdot 255$, where r and c are the numbers of grid rows and columns respectively. The resulting feature vectors are directly used for comparison in [2]. However, they can be also further processed with PCA and subsequent z-score normalization.

2.7 Multi-algorithmic Score-level Fusion

In the previous sections, the face alignment, image data extraction, image filters, iso-geodesic curves and subspace projections were described. While the face alignment is common for all recognition pipelines, image data, filters, and subspace projections are variables. One example of a possible recognition algorithm (*unit* in further text) is in Figure 2.14.

2.7.1 Score normalization

One of the most important concerns, when the score-level fusion is involved, is the score normalization [31, 33]. Comparison scores of individual units have to be normalized to

some common domain prior to fusion itself.

There are several techniques of score normalization. Although the metric should be defined so that it satisfies the non-negativity axiom ($d(x, y) \geq 0$), a biometric comparison score s as well as the normalized value s' may be lower than zero. Let S is the set of all comparison scores from some evaluation run. S_{gen} and S_{imp} are sets of all genuine and impostor scores respectively ($S = S_{gen} \cup S_{imp}, S_{gen} \cap S_{imp} = \emptyset$).

Probably the simplest normalization is min-max:

$$s' = \frac{s - \min(S)}{\max(S) - \min(S)} \tag{2.8}$$

Min-max normalization is highly sensitive to outliers. Let $\overline{s_{imp}}$ and $\overline{s_{gen}}$ denote the mean impostor and genuine scores respectively. The normalized score s' from input score s is computed using the following formula:

$$s' = \frac{s - \overline{s_{gen}}}{\overline{s_{imp}} - \overline{s_{gen}}} \tag{2.9}$$

The normalization from Equation 2.9 transforms the input score, such that the mean impostor comparison value is 1 and mean genuine comparison value is 0. Further robustness against outliers may be achieved when the median instead of mean is used:

$$s' = \frac{s - \text{median}(S_{gen})}{\text{median}(S_{imp}) - \text{median}(S_{gen})} \tag{2.10}$$

Another frequently used normalization technique is z-score. Let σ and $\overline{s}$ denote standard deviation and mean value of S respectively:

$$s' = \frac{s - \overline{s}}{\sigma} \tag{2.11}$$

The similar normalization technique is based on the *Median of Absolute Deviation* (MAD) [34]. MAD of set S is defined as:

$$\text{MAD}(S) = \text{median}\{|s_1 - \text{median}\,(S)\,|, |s_2 - \text{median}\,(S)\,|, \ldots, |s_n - \text{median}\,(S)\,|\} \tag{2.12}$$

The MAD-based normalization uses the computed median and MAD of S:

$$s' = \frac{s - \text{median}\,(S)}{\text{MAD}(S)} \tag{2.13}$$

The normalization based on the hyperbolic tangent is defined as:

$$s' = \frac{1}{2}\left(\tanh\left(\frac{s - \overline{s}}{100\sigma}\right) + 1\right) \tag{2.14}$$

2.7.2 Classifier-based fusion

Suppose that we have n recognition units. Each unit employs its own image processing, feature extraction using some subspace projection and comparison metrics. The resulting comparison scores provided by individual units are normalized using Equation 2.9. The task is to combine normalized scores to a single value that can be thresholded in order to decide whether an input scan is accepted or not.

From the machine-learning perspective, the task is to create a classifier C that is capable to assign a class label $c \in \{gen, imp\}$ (genuine or impostor) for a given vector of normalized scores $\mathbf{s} = (s_1, s_2, \ldots, s_n)$:

$$C : \mathbf{s} \mapsto c \tag{2.15}$$

In order to have both the biometric security and user convenience configurable, the vector of scores is mapped to genuine likelihood or signed distance from the genuine/impostor decision hyperplane rather than class label.

Logistic Regression

The learning of fusion based on logistic regression requires preprocessing of the training data. First, the design matrix $\mathbf{\Phi}$ is created:

$$\mathbf{\Phi} = \begin{pmatrix} 1 & \mathbf{s}_1 \\ \vdots & \vdots \\ 1 & \mathbf{s}_n \end{pmatrix} \tag{2.16}$$

Each row i in $\mathbf{\Phi}$ contains 1 in the first column followed by individual normalized comparison scores from i^{th} comparison.

The target column vector $\mathbf{t}$ contains labels corresponding to individual comparisons: $\mathbf{t} = (c_1, c_2, \ldots, c_n)^T$. c_i is set to 1 if the same (genuine) users were compared. In case of different (impostor) comparison holds $c_i = 0$.

The projection matrix $\mathbf{W}$ is computed after that:

$$\mathbf{W} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{t} \tag{2.17}$$

The genuine/impostor classification of input normalized scores $\mathbf{s}$ is based on the fusion score gained from the following equation:

$$s = \frac{1}{1 + \exp(-\mathbf{W}^T\psi)} \tag{2.18}$$

where $\psi = (1\ \mathbf{s})$.

Support Vector Machine (SVM)

In fact, Support Vector Machine is an optimization problem. SVM attempts to find a hyperplane that divides the two classes with the largest margin. The support vectors are the points which fall within this margin. If the classes are not linearly separable, soft margin SVM is introduced. Parameter C controls the number of points that may stray over the line into the margin.

In this work, the implementation libSVM [6] is used.

Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis [3] is used as the binary classifier in this case. The n-dimensional space, where n is the number of involved units, is projected to one-dimensional line that separates genuine and impostor scores clusters the best.

2.7.3 Hill-climbing Unit Selection

Beside the score normalization, another issue that has to be taken into account is to measure the score correlation and performance bias [32, 15]. If the score correlation between the involved units is high, the resulting recognition performance of the multi-algorithmic system may not be significantly better than the individual units. Moreover, huge performance bias between units may reduce the recognition performance.

The task is to select from the given set of units $U = (u_1, u_2, \ldots, u_n)$ a non-empty subset S that achieves possibly the best recognition performance. The exhaustive search of the relatively small set U is simple. However, when the number of units exceeds certain thresholds, the exhaustive search is impossible.

We employ a wrapper selection [14]. This approach has been originally developed for feature selection, however, it can be also used for the unit selection in the multi-algorithmic fusion.

The optimization criterion is achieved EER (EER_i) of a particular unit i. The algorithm is as follows:

1. Select the unit b that achieves the best EER_b, remove it from the set U and add it to the set of selected units S.
2. For each remaining unit j in U measure fusion EER of the set $S \cup j$.
3. Select the best unit and add it to S. If there was no improvement of fusion EER, exit. Otherwise return to 2.

There is a potential drawback of the hill-climbing selection – selected units might be too specific for the training set and cannot generalize. Validation on the separate set is therefore recommended. However, our further experiments did not reveal any significant performance drop between the training and testing data. It has emerged that the hill-climbing is a good choice for recognition unit selection.

Chapter 3

Evaluation

The presented algorithm will be evaluated in this chapter. In order to compare the achieved results with other available state-of-the-art algorithms, evaluation is performed on the Face Recognition Grand Challenge version 2.0. Detailed methodology, tests as well as achieved results are described in the following sections.

The face recognition algorithm was also tested on the databases obtained with low-cost 3D sensors – such are Microsoft Kinect[1] and SoftKinetic DepthSense DS325[2]. The expansion of personal depth sensors related with the new ways of the human-computer interaction in recent years markedly lowered the price of 3D acquiring devices for personal use. However, the biggest challenge of the face recognition based on the low-cost depth sensors is the quality of the acquired scans. While, for example, the Minolta Vivid or Artec 3D M scanners provide a highly precise geometry with an outstanding resolution and level of detail, the scans retrieved from the Kinect or DepthSense DS325 sensors are noisy, have low resolution and sometimes contain holes. The last two sections of this chapter are dedicated to the performance evaluation on those low-cost depth sensors.

3.1 Database Description and Evaluation Methodology

The proposed face recognition algorithm was trained and tested on the Face Recognition Grand Challenge Database v 2.0 - a standard evaluation database for facial biometrics [29]. This database contains scans captured in spring 2003, fall 2003, and spring 2004. The Spring 2004 portion (2,114 scans) was divided into five parts, such than each part contains the same count of subjects. No subject is present in more than one part. The face alignment algorithm presented in the next section failed 36 times and thus these scans were removed. See Table 3.1 for the detailed information about dividing the Spring 2004 portion of the FRGC v2.0 database for evaluation purposes.

The first part of Spring 2004 was used for the training of the face alignment and the training of individual parameters of subspace projections. The second part was used for the validation of selected parameters and for the training of final fusion. The last three parts were used for evaluation purposes.

[1] http://www.xbox.com/kinect/

[2] http://www.softkinetic.com/products/depthsensecameras.aspx

Table 3.1: FRGC2 Spring 2004 statistics.

Part	Subjects	Scans	Purpose
1	69	416	training
2	69	451	training, validating
3	69	414	evaluation
4	69	417	evaluation
5	69	380	evaluation

Table 3.2: Comparison of two face alignment approaches.

Align approach	EER
Landmark-based	0.060
ICP	0.043

3.2 Face Alignment

The alignment of the input face mesh is one of the most important tasks in the pre-processing part of the recognition pipeline. There are two main evaluation characteristics - precision and speed. The first mentioned can be implicitly evaluated by the evaluation of the overall biometric performance. The latter is important for practical purposes.

The alignment of the face can be performed in several ways. It can rely on the landmark detection. The inner corner of the eyes, nose tip, and nose corners are detected and the face is aligned subsequently such that the sum of absolute differences between landmarks on the input face and landmarks on the reference face template is minimal. Although the alignment of the predefined points is simple and fast to compute, the notable downside of this approach is the requirement of precise landmark detection. Just a slight inaccuracy can lead to a wrong alignment, and the recognition performance is thus negatively affected.

The second option of the face alignment is the involvement of *Iterative Closest Point* (ICP) algorithm. There is no initial landmark estimation. The entire face mesh model is aligned to the reference mean face template. However, this approach consumes much more CPU power.

Both alignment approaches were evaluated on the FRGC database. The first part of the Spring 2004 set was used for training and the second part for validation. The recognition was performed on the range images of size 120×120 pixels. Features were extracted using PCA and individual feature vector components subsequently normalized using z-score. The correlation metric was employed for comparing the resulting feature vectors. The evaluation results of both approaches are in Table 3.2.

The alignment utilizing the ICP algorithm outperformed landmark-based approach. Therefore it will be used in the following test. On the other hand, the drawback of the ICP is much higher CPU usage. The problem comes from the fact that there is no explicit mapping between the input face mesh and the mean reference template. In each ICP iteration, the points between the input and the reference have to be associated. This can be speeded-up when the index (e.g. k-means) of the points from the input face is created. However, the points in each iteration are changing their position and the index has to be recalculated. This issue might be solved with the inverse transformation trick described previously in Section 2.2.2. The comparison of mean duration of ICP alignment for FRGC scans is in Figure 3.1.

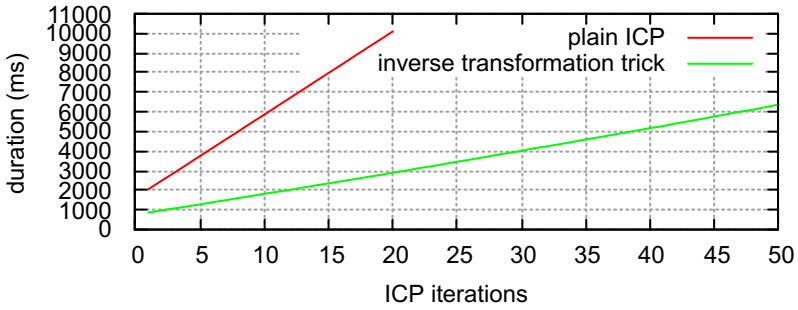
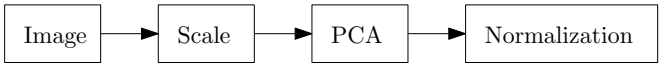


Figure 3.1: Comparison of ICP alignment approaches for FRGC scans. The duration (in milliseconds) is drawn for plain ICP as well as for ICP alignment with the inverse transformation trick.

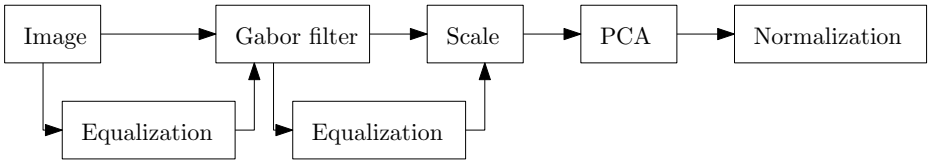
3.3 Evaluation of Individual Recognition Units

The individual recognition units were evaluated. Each unit is represented by input image data (e.g. texture, depth or curvature representation) on which some filters are applied. The resulting filter response is further processed with some feature extraction technique (e.g. PCA). The following types of units were tested:

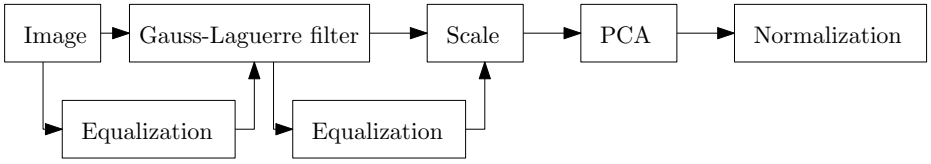
Plain image units – the input image (texture, depth, curvature) is scaled to half of its size, processed with PCA subspace projection and z-score normalized. Individual feature vectors are compared with correlation metric.



Gabor-based image units – the input image is processed by Gabor filter with a specific size and orientation, scaled to half size, projected using PCA and normalized. Correlation metric is used. Image is optionally equalized before or after the Gabor filter application.



Gauss-Laguerre-based image units are similar to Gabor-based units, except that the Gauss-Laguerre filter is used.



LBP units – the input image is equalized optionally. After that, it is blurred slightly with Gauss filter. Finally, the LBP filter is applied, the image is scaled to half of its size and projected using PCA. Z-score normalized feature vectors are compared with the correlation metric.

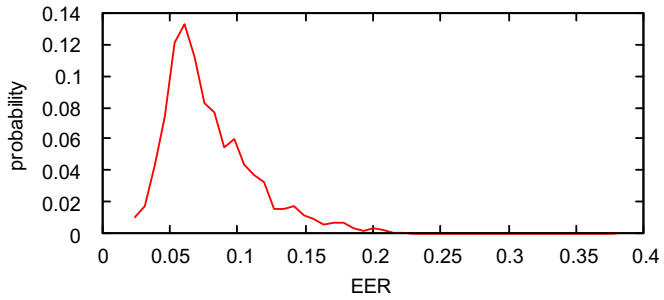
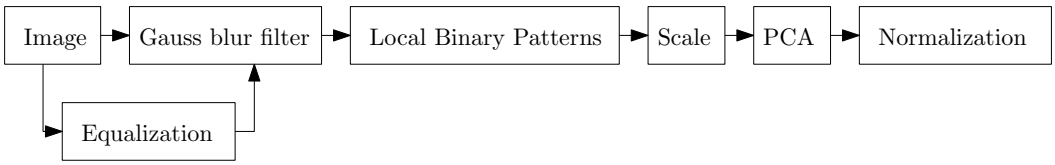


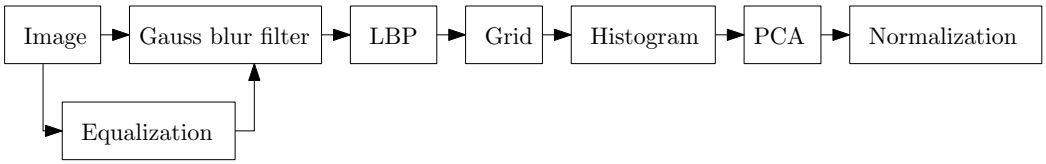
Figure 3.2: Histogram of EERs for all tested recognition units.

Table 3.3: Partial results of the units evaluation - the best representatives of each unit type.

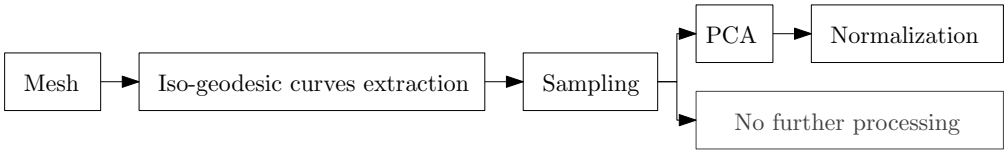
Rank	Type	Input data	Applied filters	EER
1	LBP histogram	Gaussian curvature	gaussBlur(7); LBP; histogram(10,9)	0.0247
7	Gauss-Laguerre	Range image	gaussLag(48,1,0); scale(0.5)	0.0267
12	Gabor	Eigencurvature	gaborAbs(1,2); equalize; scale(0.5)	0.0295
34	LBP	Mean curvature	gaussBlur(11); LBP; scale(0.5)	0.0361
73	Plain	Range image	scale(0.5)	0.0416
759	Iso-geodesic curves		5 curves centered at the nosetip	0.0769



LBP units with histogram are similar to the plain LBP units but the resulting image after the LBP application is divided into a grid and the histogram of intensity values is calculated in each cell of the grid. The resulting histograms are concatenated and further processed with PCA.



Iso-geodesic curves – 5 iso-geodesic curves are extracted at a specific point with geodesic distance 1, 2, 3, 4, and 5 cm. Curves are sampled to 100 points per curve. After that, coordinates of individual points are concatenated to one column vector. It is compared with other feature vector using the city-block metric or processed with PCA and z-score normalized. The correlation metric is used in the second case.



There were evaluated 1,720 different units. The histogram of achieved equal error rates for each of them is in Figure 3.2. Partial results are in Table 3.3

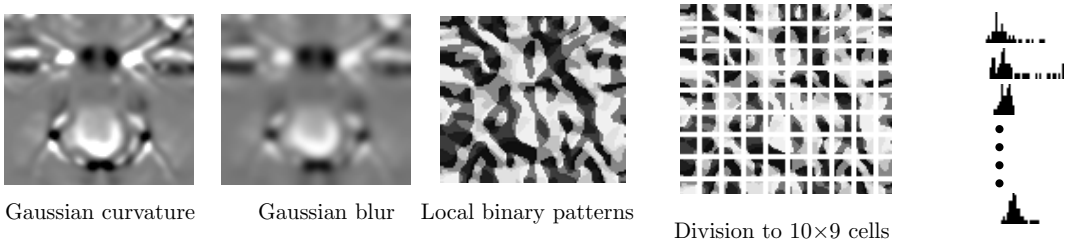


Figure 3.3: LBP-based recognition unit. The input Gaussian curvature image is blurred using Gaussian kernel. After that, local binary patterns are calculated. The image is divided into a grid and a histogram of intensity values is calculated within each grid cell. Individual histogram values are concatenated and further projected using PCA. The feature vector is thus created.

The best unit consists of the following pipeline. The Gaussian curvature representation of the surface is blurred with the Gaussian kernel of size 11 pixels. This image is further processed with a local binary patterns filter. The resulting image is divided into 10 horizontal and 9 vertical cells. A histogram of values within each cell of size 10×10 pixels is calculated. A set of histograms is further processed with PCA in order to reduce the correlation of values as well as the number of feature vector components. The entire process is depicted in Figure 3.3.

Another example is 7^{th} best unit. A range image (depthmap) is convolved with both real and imaginary Gauss-Laguerre kernels of size 48 pixels and parameters set to $n = 1$ and $k = 0$. The absolute response is calculated from real and imaginary responses. The resulting image is scaled with factor 0.5 to size 50×45 pixels and finally processed with PCA.

The 12^{th} best unit applies Gabor filter with scale 1 and orientation 2 on the eigencurvature surface representation. The histogram of absolute response is equalized and scaled with factor 0.5. As in the previous cases, PCA is applied on the image to reduce the feature vector size.

3.4 Multi-algorithmic Fusion

3.4.1 Score Normalization Techniques

Score-normalization is an important task preceding the fusion itself. Individual comparison scores have to be transformed into common domain in order to combine them meaningfully.

Figure 3.4 shows the score normalization result of three different recognition units. Each unit employs specific comparison metric and thus the score ranges vary. Every graph shows an impostor-genuine distribution of the comparison scores. The red curve belongs to the unit employing the correlation metric, the green curve corresponds to the unit with the Euclidean comparison and the blue curve corresponds to the unit with the city-block comparison metric. The graph in the first row shows the original pre-normalized values. Each subsequent graph shows normalized values where the normalization was achieved using one of the techniques previously described in Section 2.7.1.

The good normalization technique is able to align the curves of the impostor-genuine distribution that comes from different recognition units having feature vectors of different

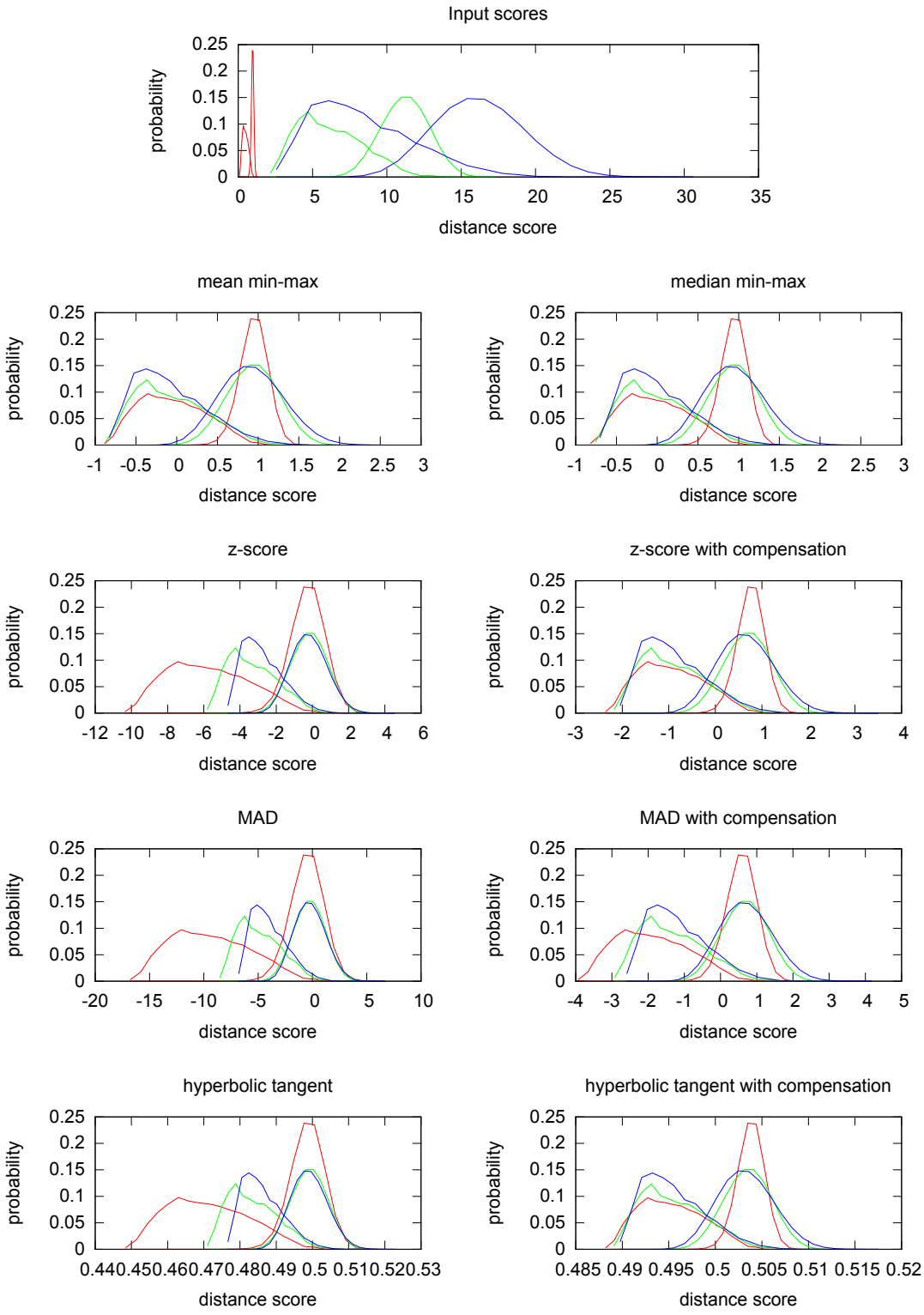


Figure 3.4: Score normalization techniques - genuine/impostor score distributions for unis employing correlation metric (red), city-block metric (green) and Euclidean distance (blue).

Table 3.4: Hill-climbing selection of individual units for SVM-based score-level fusion classifier.

Iteration	Unit		Unit EER	Fusion EER
	Input data	Filters		
1	shape index	GaussBlur(7); LBP; histogram(10,9)	0.0247	0.0247
2	range image	GaussLag(48,1)	0.0266	0.0114
3	texture	Equalize(); GaussLag(64, 4)	0.0963	0.0075
4	shape index	Gabor(3, 2); Equalize()	0.0585	0.0063
5	mean curvature	Gabor(2, 4); Equalize()	0.0657	0.0050
6	texture	Gabor(3, 2)	0.1391	0.0043
7	eigencurvature	Gabor(4, 2); Equalize()	0.0522	0.0035
8	shape index	Equalize(); GaussLag(48, 5)	0.0881	0.0030
9		iso-geodesic curves	0.1163	0.0028
10	range image	Equalize(); Gabor(7, 7)	0.1079	0.0022
11	Gaussian curvature	Equalize(); GaussLag(64, 1)	0.0564	0.00175
12	mean curvature	Gabor(7, 0); Equalize();	0.1014	0.00174
13	eigencurvature	Gabor(1, 0)	0.0642	0.00173

dimensionality and employing various metrics. It can be also measured implicitly by evaluation of the overall biometric performance. It has emerged that simple mean min-max normalization (see Equation 2.9) is the best choice for our purposes.

3.4.2 Greedy Hill-climbing Unit Selection

The greedy hill-climbing unit selection for final fusion was described previously in Chapter 2.7.3. All units from Section 3.3 were used as an input to the hill-climbing selector. The score-level fusion was provided by binary SVM classifier. Another classifiers, density-based and combination techniques were used as well in further experiments.

The hill-climbing selector chose 13 units – see Table 3.4. In the first iteration, the unit employing the application of LBP histogram on the shape index image was selected. The subsequent iteration chose a specific Gauss-Laguerre filter applied on the range (depth) image. The equalized texture followed by the application of Gauss-Laguerre filter was selected in the third iteration.

3.4.3 Comparison of Fusion Techniques

In the subsequent experiment, individual fusion techniques were compared. The selection of 13 units gained by the hill-climbing was used as the input to the training of multi-algorithmic fusion. Moreover, new scans were evaluated in order to test robustness of the fusion techniques. The results are shown in Table 3.5.

The transformation-based fusion is represented by a simple sum rule and the weighted sum. The weights of individual units are proportional to the achieved EER on the training set. For example, if the EER of the unit i is e_i, the corresponding weight is set to $w = 0.5 - e_i$.

The only representative of the density-based fusion is *Gaussian Mixture Model* (GMM). The probability density distributions of impostor as well as the genuine scores were modeled using 5 Gaussians with diagonal covariance matrices.

The classifier-based fusion is represented by logistic regression, LDA, and SVM. A plain linear kernel was used. The experiments suggest that there is no significant difference in recognition performance between the individual fusion techniques.

Table 3.5: Comparison of individual score-level fusion techniques (evaluated on the 3rd part of Spring 2004 portion).

Fusion technique	EER on training set	EER on testing set	FNMR at given FMR 0.001	0.0001	0.00001
sum	0.0096	0.0106	0.0459	0.0968	0.1838
weighted sum	0.0091	0.0106	0.0424	0.0954	0.1824
GMM	0.0050	0.0099	0.0353	0.0600	0.1497
LogR	0.0069	0.0112	0.0332	0.0912	0.1478
LDA	0.0083	0.0105	0.0424	0.0968	0.1669
SVM	0.0017	0.0096	0.0388	0.0721	0.1258

Table 3.6: Achieved results on the Spring 2004 evaluation subset.

Evaluation partition	FNMR at given FMR 0.01	0.001	0.0001
1	0.0091	0.0388	0.0721
2	0.0098	0.0329	0.0596
3	0.0167	0.0589	0.1091
Median	0.0098	0.0388	0.0721

3.5 Comparison with the State-of-the-art

The results of Face Recognition Vendor Test 2006 (FRVT) was based on the evaluation of almost the entire FRGC dataset [28]. The performance of a biometric system vary with different sets of biometric samples. It is important to measure both the overall performance of a biometric system and the scale of the variability to measure statistical uncertainty. In the FRVT, the performance variability is measured by partitioning the test images into a set of smaller test sets. The performance is then computed on each of the partitions. According to the FRVT report, 3,589 out of the 5,000 scans in FRGC were used for the evaluation. These scans were divided into 13 partitions with the total count of 330 subjects. Unfortunately, the selection of these 3,598 scans is not clear from the report. Therefore, we bring the comparison of best algorithms involved in FRVT with our achieved results reduced only to Spring 2004 part in this section.

We have utilized the Spring 2004 FRGC subset such that its evaluation part contains 1211 scans from 207 individuals designated for the evaluation. This evaluation subset was divided into 3 partitions, as it was mentioned earlier in Section 3.1. Table 3.6 shows achieved results with a classifier utilizing SVM-based fusion. The comparison with other FRVT competitors is in Table 3.7. While the Viisage algorithm outperforms all others, our algorithm achieves the second best results.

However, the presented comparison is for illustration purposes only, since we did not follow the same evaluation methodology as in the FRVT. One of the goals of this thesis is the utilization of low-cost depth sensors for the 3D face recognition. The comprehensive evaluation on databases obtained with SoftKinetic DS325 and Microsoft Kinect 360 is presented in the next two sections.

Table 3.7: Comparison of our method with FRVT competitors. The exact numbers were taken from the graphs in appendix section of FRVT report [28].

Algorithm		Median FNMR at given FMR		
Name / Organization	Abbrev.	0.01	0.001	0.0001
Cognitec	Cog1	0.050	0.070	0.160
Geometrix	Geo	0.035	0.085	0.155
Univ. of Houston	Ho1	0.030	0.050	0.100
	Ho3	0.025	0.050	0.095
Tsinghua Univ.	Ts1	0.035	0.145	-
Viisage	V	0.005	0.020	0.070
	Va	0.010	0.055	0.170
Our method		0.010	0.039	0.072

Figure 3.5: Example of scans in the SoftKinetic database (processed, aligned, and cropped).

3.6 Evaluation on SoftKinetic Database

We have created the SoftKinetic database during spring 2014. It contains 398 scans from 52 individuals. During the capturing, the emphasis was put on following points:

- Various lighting conditions.
- Various (but limited) facial expressions – we allowed the subjects to have a slight smile, lifted eyebrows or frowned face.
- Scanning of some subject was splitted into several sessions in different days.
- An effort was made to have a diversity in gender, race, and age of scanned subjects.

The example of some scans in the SoftKinetic database is in Figure 3.5. Contrary to the scans acquired with the Minolta Vivid scanner (FRGC database), the data captured with SoftKinetic DepthSense DS325 sensor suffer from high noise among the z axis [17, 8], therefore some sort of denoising has to be applied on the 3D models in the preprocessing portion of the recognition pipeline.

Although one can use a stronger Gaussian smooth filter, our experiments show that much better, in terms of recognition performance, is the application of the feature-preserving mesh denoising algorithm [38]. An example of application of such filter is in Figure 3.6.

The field of view of DS325 sensor is very wide ($74° \times 58° \times 87°$)[3]. Therefore, the quality and resolution of face scans rapidly decreases when the subject moves away from the sensor. Two scans from the same subject acquired from the distance of 35cm and 70cm are depicted

[3] http://www.softkinetic.com/en-us/products/depthsensecameras.aspx

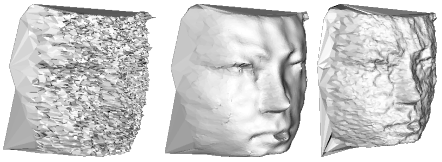


Figure 3.6: Application of feature preserving mesh denoising – before (left) and after (middle). Basic Gaussian smoothing is on the right side of the figure.

in Figure 3.7. While the near-scan contains 6,951 verticies, the far scan contains only 1,560 verticies, which is more than 4 times fewer.

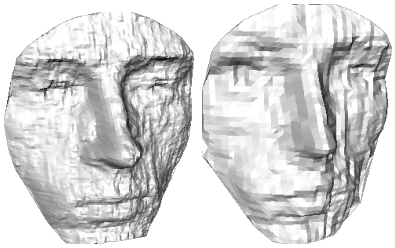


Figure 3.7: Two scans acquired with the SoftKinetic DS325 sensor captured from 35cm (left) and 70cm (right).

3.6.1 Finding Suitable Smoothing and Denoising Algorithm

The initial test evaluated recognition performance on the range images and shape index images. The scans from the training part of the SoftKinetic dataset were subsequently smoothed and aligned using ICP algorithm. After that, range images and shape index images were calculated. PCA trained on FRGC depth and shape index images with zScore normalization was used in order to extract features. Feature vectors were compared using correlation metric.

Both Gaussian smooth filter (Gauss) and feature-preserving mesh denoising (M-Denoise) algorithms with various parameters were evaluated. The results are in Table 3.8. The lowest EER was achieved when the M-Denoise filter was applied.

3.6.2 Multi-Algorithmic Fusion

The multi-algorithmic fusion similar to the fusion used for FRGC (see Section 3.4) was trained and evaluated. The hill-climbing optimization selected 10 units. Contrary to the FRGC fusion, no iso-geodesic or LBP-based recognition unit is present. The lower quality of SoftKinetic scans causes the absence of the iso-geodesic curves unit. On the other hand, missing LBP-based unit is not so obvious. The selected units are in Table 3.9.

The DET curves from the evaluation of the training as well as test parts of the SoftKinetic dataset are in Figure 3.8. The impostor-genuine score distributions are in Figure 3.9 and the particular achieved FNMRs at given FMRs are in Table 3.10.

Table 3.8: Evaluation of mesh smoothing and denosing algorithms on the SoftKinetic dataset.

Smoothing method	Smooth iterations	Smoothing parameter	EER on range images	EER on shape index images
None	-	-	0.077	0.135
M-Denoise	5	0.01	0.053	0.116
M-Denoise	5	0.02	0.052	0.114
M-Denoise	5	0.04	0.052	0.108
M-Denoise	10	0.01	0.044	0.111
M-Denoise	10	0.02	0.048	0.103
M-Denoise	10	0.04	0.046	0.105
M-Denoise	20	0.01	0.041	0.097
M-Denoise	**20**	**0.02**	**0.041**	**0.095**
M-Denoise	**20**	**0.04**	**0.040**	**0.096**
Z-Smooth	5	0.2	0.068	0.131
Z-Smooth	5	0.5	0.061	0.120
Z-Smooth	5	1.0	0.057	0.114
Z-Smooth	10	0.2	0.067	0.119
Z-Smooth	10	0.5	0.061	0.119
Z-Smooth	10	1.0	0.052	0.110
Z-Smooth	20	0.2	0.061	0.121
Z-Smooth	20	0.5	0.053	0.119
Z-Smooth	20	1.0	0.042	0.111

Table 3.9: Selected recognition units gained from the training part of the SoftKinetic dataset.

Unit	Input data	Filters
1	range image	GaussLag(72, 2)
2	texture	Equalize(); Gabor(5, 6)
3	range image	Gabor(3, 1)
4	texture	Equalize(); Gabor(6, 0);
5	eigencurvature	GaussLag(24, 5)
6	texture	Gabor(4, 2)
7	range image	GaussLag(48, 0);
8	shape index	GaussLag(96, 1)
9	shape index	Gabor(2, 5)
10	texture	Equalize(); Gabor(2, 7)

Table 3.10: Evaluation on SoftKinetic database results.

Data	EER	FNMR at given FMR		
		0.01	0.001	0.0001
Train set	0.011	0.012	0.043	0.098
Test set	0.043	0.087	0.192	0.312

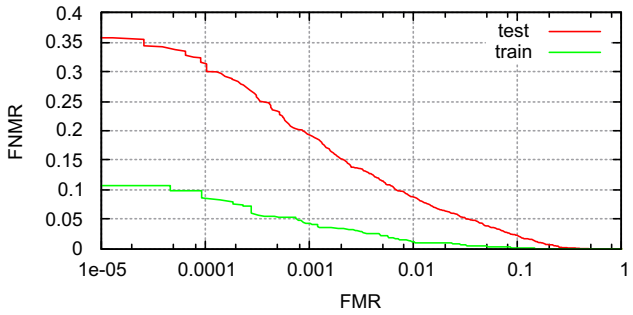


Figure 3.8: DET curves from the SoftKinetic evaluation.

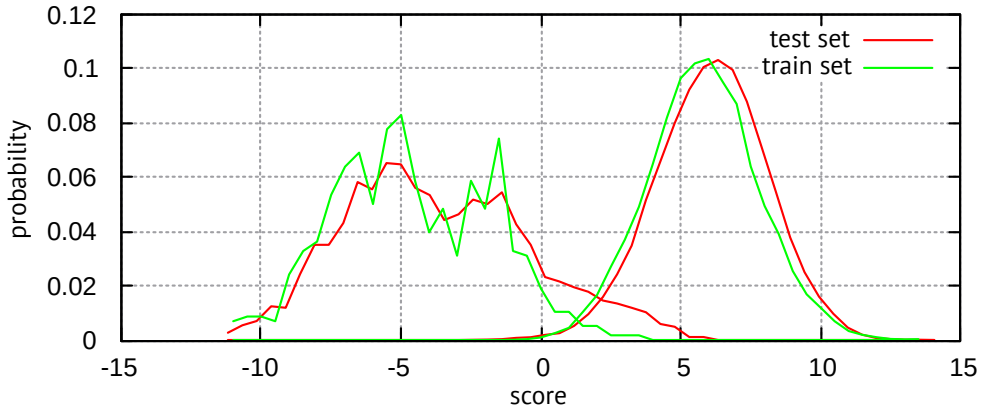


Figure 3.9: Impostor-genuine score distributions from the SoftKinetic evaluation.

Figure 3.10: Scans of one subject from all sessions in SoftKinetic dataset.

3.6.3 Real-World Scenarios

The real-world implementation of biometric system has to deal with the problems that are not always considered when it is being evaluated in the laboratory conditions. Usually, the users are not experienced enough in order to position their face such that the biometric system achieves the best results. The lighting conditions vary and the mood of users affecting their 3D face appearance also changes. All these factors can decrease the biometric performance rapidly.

The most convenient scenario for the users of an access-control biometric system is the identification. Users do not need to claim their identity before the scanning process begins. Based on their identity, the system decides if they have an authorization to proceed.

In this subsection, the template aging and identification with respect to the real-world application will be evaluated. The enrollment of a new user to the biometric system usually consists of acquiring several (four) scans for the creation of a reference template. The problem is that these scans are acquired in very short time period (just few seconds) and the lighting conditions are still the same. Moreover, the users use the biometric system for the first time, they are fully concentrated on the capture process and thus they have no facial expressions. When they use the biometric system later, the environment condition might change as well as the user's mood might affect mimics of the face.

Subjects with more than or equal of 15 scans were selected from the SoftKinetic dataset. When the dataset was captured, 5 scans were acquired in each session. This means that if the subject participated in 4 sessions, 20 scans of this particular subject are stored in the dataset. The sessions took place in different days in different places. The example of all scans of one subject from all sessions is shown in Figure 3.10.

The first four scans of each subject were used for creating of the gallery templates. The remaining scans were used for the evaluation. When the input probe is compared with the gallery templates, the arithmetic mean of individual comparisons between the probe and the templates is returned. The time evolution of the comparison scores is in Figure 3.11. If the decision threshold is set to 0 or lower, there will be just one false reject – last scan of subject `subj4`.

Once we have more than one reference template (4 in this case), we can have several decision strategies to compare the input feature vector with the stored templates. The simplest approach is a $1 : N$ comparison. The resulting identity is based on the reference template with the smallest distance between the input feature vector i and template t_i, where $i \in \{1, 2, \ldots, N\}$ (see Figure 3.12 left). The more robust method that can handle outlying reference template is in the same figure on the right. The classification is based on the average distance to the k nearest reference templates for class C_j. Let the $\min_k S$ denote the k minimal elements in set S. The distance between input vector X and class

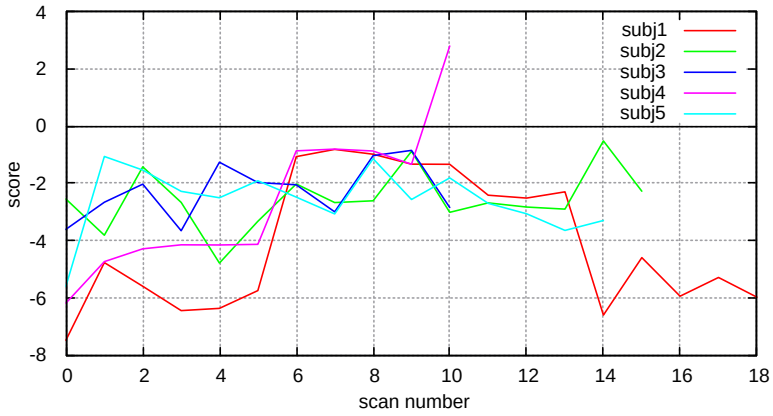
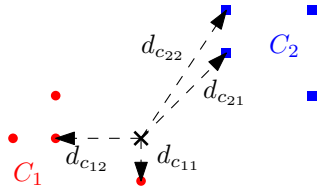


Figure 3.11: Time evolution of the comparison scores from SoftKinetic dataset subjects with more than or equal of 15 scans. Except for the last scan of the subject 4, all other scans were successfully verified with a comparison score lower than zero.



$$d(X, C_1) = \min(d_{c_{11}}, d_{c_{12}}, \ldots, d_{c_{1j}}) \qquad d(X, C_1) = \frac{d_{c_{11}} + d_{c_{12}}}{2}$$

$$d(X, C_2) = \min(d_{c_{21}}, d_{c_{22}}, \ldots, d_{c_{2k}}) \qquad d(X, C_2) = \frac{d_{c_{21}} + d_{c_{22}}}{2}$$

Figure 3.12: Classification strategies for identification in biometric systems. Basic $1 : N$ comparison is on the left, more robust approach is on the right hand side.

$C_j = \{c_{j_1}, c_{j_2}, \ldots, c_{j_m}\}$ is:

$$d(X, C_j) = \frac{\sum \min_k \{d(X, c_{j_1}), d(X, c_{j_2}), \ldots, d(X, c_{j_m})\}}{k} \tag{3.1}$$

The question is, if the system employing the modified distance metric from Equation 3.1 outperforms basic $1 : N$ comparison. The results are in Table 3.11. There are two approaches for the creation of the reference templates. The first (denoted *without randomizing*) was described previously – the first 4 scans from each subject were used for template, remaining scans were used for the evaluation. The latter approach (denoted *with randomizing*) randomly selects 4 scans for the template creation among all subject scans. The results are a bit surprising. The modified distance metric does not outperform the simple $1 : N$ comparison. Moreover, it is worse, when the randomized templates are used. On the other hand, the randomized templates outperform the non-randomized significantly. This was, however, expected, as the randomized templates capture more intra-class variability.

3.7 Kinect Evaluation

We would like to show that the proposed algorithm is robust enough in order to be easily adapted to any depth sensor. Kinect cannot be power supplied by a USB cable and re-

Table 3.11: Identification evaluation on SoftKinetic dataset.

Reference template count (k)	without randomizing		with randomizing	
	EER	FNMR @ FMR = 0.01	EER	FNMR @ FMR = 0.01
1	0.147	**0.278**	**0.028**	**0.052**
2	0.152	0.310	0.052	0.084
3	**0.142**	0.321	0.063	0.121
4	**0.142**	0.310	0.073	0.136

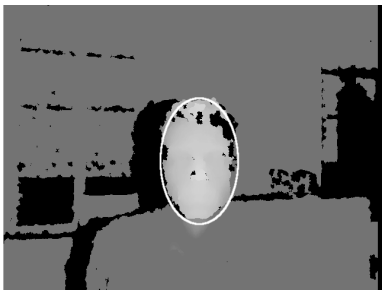


Figure 3.13: Range image of the scanned face. Face within the image is marked with the white ellipse.

quires an external power adapter. This restricts its usage in the embedded face recognition systems. On the other hand, since it is quite often used in households, the utilization of a 3D face recognition is shifted from security applications to home entertainment.

Contrary to the SoftKinetic DepthSense DS325, Kinect sensor is designated for scanning of the entire room and full-body capturing. The minimal distance where the depth is captured is 80cm, but the practical limit for capturing is rather 120cm. Moreover, the fields of view of both the depth sensor and the RGB sensor are very wide such that almost entire room is captured. When a subject is scanned with Kinect, just 10% of sensor is used. On the other hand, Kinect scans require less denoising treatment than those captured with SoftKinetic DS325. Therefore, the biggest challenge of the Kinect dataset is the small resolution of the input meshes rather then the noise. For an example of Kinect range visualization with a marked face region see Figure 3.13. An average Kinect face mesh contains about 5,000 vertices.

Our Kinect database consists of 108 scans divided into two parts - training and testing. Each part contains 9 different subjects that provided 6 scans. Facial expressions as well as varying lighting conditions are present in some scans. The example of Kinect database is in Figure 3.14.

The DET curve of our recognition algorithm evaluated on the Kinect database is in Figure 3.15. SVM classifier has been used for the final fusion of the individual recognition

Figure 3.14: Some examples from the Kinect database.

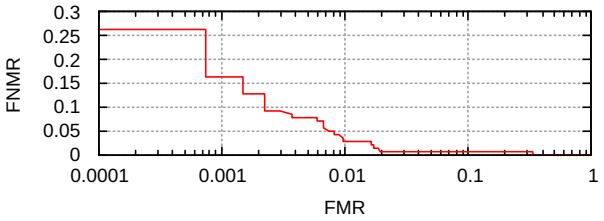


Figure 3.15: Evaluation of SVM fusion on the Kinect database – DET curve.

Table 3.12: Hill-climbing selection of individual units for SVM-based score-level fusion classifier on the Kinect database.

Iteration	Unit		Unit EER	Fusion EER
	Input data	Filters		
1	texture	DoG(5, 3); GaussBlur(11); LBP; histogram(10,9)	0.0639	0.0639
2	range image	GaussLag(64, 2)	0.0983	0.0337
3	eigencurvature	Gabor(5, 5)	0.3191	0.0210
4	mean curvature	Gabor(2, 6)	0.3601	0.0152
5	mean curvature	Gabor(6, 5)	0.3555	0.0137
6	shape index	GaussLag(64, 2)	0.2778	0.0134

units. The process of selecting the individual units is in Table 3.12. Iso-geodesic curves were not selected for the final fusion. This is probably due to the fact that the Kinect scans are quite rough and thus the curves do not contain much discriminative ability. On the other hand, a unit utilizing texture images processed with the *Difference of Gaussians* (DoG) filter was selected in the first place.

Chapter 4

Conclusion

This thesis presented a 3D face recognition approach based on the multi-algorithmic fusion of individual units utilizing iso-geodesic curves and specific image filters. The hill-climbing selection was used in order to combine only those units that have a positive impact on the recognition performance.

The idea of the multi-algorithmic approach has been published in [20, 25, 21]. These two papers and the book present the combination of anatomical soft-biometrics and holistic algorithms. It shows that the combination of multiple algorithms improves the recognition performance. We have utilized biometrics fusion in [26]. This paper describes the thermal face recognition pipeline where multiple subspace projection techniques are combined. The further extension of this approach has been presented in [42]. We have added the image filters prior to the subspace projections. The overview of the thermal face as well as 3D face recognition techniques was described in chapter "3D and Thermo-face Recognition" of the book "New Trends and Developments in Biometrics" [22].

The utilization of the hill-climbing unit selection was presented in [23]. This paper describes the basic idea of the iterative selection of those recognition units into a resulting multi-algorithmic system. A more robust selection has subsequently been presented in [24]. The main focus of this paper was targeting the 3D face recognition to low-cost depth sensors, such are Microsoft Kinect or SoftKinetic DepthSense DS325.

The presented recognition method requires user collaboration – the scanned subject has to be in a specific range from the sensor, look towards the camera and have a neutral face expression. All the mentioned factors (distance, dramatic facial expressions, and head rotation) can decrease the recognition performance although their impact can be reduced to some extent. The head rotation and distance from the sensor is easily compensable by the ICP registration. Facial expressions are solved implicitly - by selecting only the rigid parts of the face and selecting only recognition units robust to deformations caused by facial expressions.

The recognition algorithm was trained and evaluated on publicly available FRGC database. Moreover, we have conducted tests on our own databases acquired with Kinect and DepthSense DS325 sensors. Our results suggest that even the low-cost depth sensors that provide poor depth accuracy and noisy output can be used for successful identification in a relatively small database (up to 100 users). Our final experiments show that the main face recognition challenges - head orientation, facial mimics and varying lighting conditions may be solved. On the other hand, dioptric glasses pose difficulties. Recognition of persons wearing glasses and twins may be a promising direction for further research.

List of Abbreviations

DET – *Detection Error Trade-off* – graphical plot of error rates plotting *False Match Rates* (FMR) against *False Non-Match Rates* (FNMR).

DoG – *Difference of Gaussians* – feature enhancement filter that subtracts one blurred version of an original image from another, less blurred version of the original. See Section 2.4.3.

EER – *Equal Error Rate* is a value where the false rejection rate and false acceptance rate for a given decision threshold are equal. It is often used as criteria for evaluating performance of the biometric systems.

FRGC – *Face Recognition Grand Challenge* is a large dataset of three-dimensional face scans as well as high and low resolution photographs captured in controlled and uncontrolled lighting conditions [29].

GMM – *Gaussian Mixture Models* are formed by combining multivariate normal density components. Gaussian mixture models are often used for data clustering.

ICP – *Iterative Closest Point* algorithm minimizes the difference between two clouds of points by transforming one to the other one. See Section 2.2.2.

LBP – *Local Binary Pattern* is a type of feature used for classification in the computer vision [2].

LDA – *Linear Discriminant Analysis* is a subspace projection technique that seeks for vectors that provide the best discrimination between classes after the projection [3].

PCA – *Principal Component Analysis* is a subspace projection where the dimensionality reduction is based on the data distribution [41].

ICA – *Independent Component Analysis* is a subspace projection looking for the transformation of the input data that maximizes non-gaussianity [12].

SVM – *Support Vector Machine* is a binary classifier attempting to find a hyperplane that divides the two classes with the largest margin. See Section 2.7.2.

Bibliography

[1] H. Ahmadi and A. Pousaberi. "An Efficient Iris Coding Based on Gauss-Laguerre Wavelets". In: *Advances in Biometrics*. 2007, pp. 917–926. ISBN: 978-3-540-74548-8.

[2] T. Ahonen, A. Hadid, and M. Pietikäinen. "Face Recognition With Local Binary Patterns". In: *8th European Conference on Computer Vision*. 2004, pp. 469–481. ISBN: 978-3-540-21984-2.

[3] P. Belhumeur, J. Hespanha, and D. Kriegman. "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.7 (1997), pp. 711–720. ISSN: 0162-8828.

[4] S. Berretti, A. Del Bimbo, and P. Pala. "3D Face Recognition Using iso-Geodesic Stripes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.12 (Dec. 2010), pp. 2162–2177. ISSN: 1939-3539.

[5] P. J. Besl and H. D. McKay. "A Method for Registration of 3-D Shapes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (1992), pp. 239–256. ISSN: 0162-8828.

[6] C. Chang and C. Lin. "LIBSVM: A Library for Support Vector Machines". In: *ACM Transactions on Intelligent Systems and Technology* 2.3 (2011), p. 27. ISSN: 2157-6904.

[7] J. Daugman. "How Iris Recognition Works". In: *IEEE Transactions on Circuits and Systems for Video Technology* 14.1 (Jan. 2004), pp. 21–30. ISSN: 1051-8215.

[8] D. Falie and V. Buzuloiu. "Noise Characteristics of 3D Time-of-Flight Cameras". In: *International Symposium on Signals, Circuits and Systems*. IEEE, July 2007, p. 4. ISBN: 1-4244-0968-3.

[9] S. Feng, H. Krim, and I. A. Kogan. "3D Face Recognition Using Euclidean Integral Invariants Signature". In: *IEEE/SP 14th Workshop on Statistical Signal Processing (SSP '07)*. 2007, pp. 156–160. ISBN: 978-1-4244-1198-6.

[10] Y. Hamamoto et al. "A Gabor Filter-Based Method for Recognizing Handwritten Numerals". In: *Pattern Recognition* 31.4 (1998), pp. 395–400. ISSN: 0031-3203.

[11] L. Hong, Y. Wan, and A. Jain. "Fingerprint Image Enhancement: Algorithm and Performance Evaluation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.8 (1998), pp. 777–789. ISSN: 0162-8828.

[12] A. Hyvärinen. "Survey on Independent Component Analysis". In: *Neural Computing Surveys* 2 (1999), pp. 94–128.

[13] S. Jahanbin, R. Jahanbin, and A. C. Bovik. "Passive Three Dimensional Face Recognition Using Iso-Geodesic Contours and Procrustes Analysis". In: *International Journal of Computer Vision* 105.1 (June 2013), pp. 87–108. ISSN: 0920-5691.

[14] R Kohavi. “Wrappers for Feature Subset Selection”. In: *Artificial intelligence* 97.1 (1997), pp. 273–324. ISSN: 0004-3702.

[15] L. I. Kuncheva et al. “Is Independence Good for Combining Classifiers?” In: *Proceedings of 15th International Conference on Pattern Recognition (ICPR-2000)*. Vol. 2. IEEE, 2000, pp. 168–171. ISBN: 0-7695-0750-6.

[16] D. T. Lee and B. J. Schachter. “Two Algorithms for Constructing a Delaunay Triangulation”. In: *International Journal of Computer & Information Sciences* 9.3 (1980), pp. 219–242. ISSN: 00917036.

[17] F. Lenzen, H. Schäfer, and C. Garbe. “Denoising time-of-flight data with adaptive total variation”. In: *Lecture Notes in Computer Science*. Vol. 6938 LNCS. Springer Berlin Heidelberg, 2011, pp. 337–346. ISBN: 978-3-642-24027-0.

[18] *Machine Readable Travel Documents, ICAO doc 9303*. Tech. rep. ICAO - International Civil Aviation Organization, 2006, p. 54.

[19] M. H. Mahoor and M. Abdel-Mottaleb. “Face Recognition Based on 3D Ridge Images Obtained from Range Data”. In: *Pattern Recognition* 42.3 (2009), pp. 445–451. ISSN: 0031-3203.

[20] Š. Mráček. “3D Face Recognition”. In: *Proceedings of the 16th Conference and Competition STUDENT EEICT 2010*. Brno, CZ, 2010, pp. 124–126. ISBN: 978-80-214-4078-4.

[21] Š. Mráček. *3D Face Recognition*. Saarbrücken: Lambert Academic Publishing, 2011. ISBN: 978-3-8465-4450-1.

[22] Š. Mráček et al. “3D and Thermo-face Recognition”. In: *New Trends and Developments in Biometrics*. 2012, pp. 31–59. ISBN: 978-9-53510-859-7.

[23] Š. Mráček et al. “3D Face Recognition Based on the Hierarchical Score-Level Fusion Classifiers”. In: *Biometric and Surveillance Technology for Human and Activity Identification XI*. 2014. ISBN: 978-1-62841-012-9.

[24] Š. Mráček et al. “3D Face Recognition on Low-Cost Depth Sensors”. In: *Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG 2014)*. 2014. ISBN: 978-3-88579-624-4.

[25] Š. Mráček et al. “Inspired by Bertillon - Recognition Based on Anatomical Features from 3D Face Scans”. In: *Proceedings of the 3rd International Workshop on Security and Communication Networks*. 2011, pp. 53–58. ISBN: 978-82-91313-67-2.

[26] Š. Mráček et al. “Thermal Face Recognition - Fusion of Common Used Methods”. In: *Proceedings of the Emerging Security Technologies (EST 2012)*. 2012. ISBN: 978-0-7695-4791-6.

[27] M. Muja and D. G. Lowe. “Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration”. In: *VISAPP International Conference on Computer Vision Theory and Applications*. 2009, pp. 331–340.

[28] P. J. Phillips et al. “FRVT 2006 and ICE 2006 Large-Scale Experimental Results”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.5 (May 2010), pp. 831–46. ISSN: 1939-3539.

[29] P. J. Phillips et al. “Overview of the Face Recognition Grand Challenge”. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. IEEE, 2005, pp. 947–954. ISBN: 0-7695-2372-2.

[30] J. Pineda. “A parallel algorithm for polygon rasterization”. In: *ACM SIGGRAPH Computer Graphics* 22.4 (1988), pp. 17–20. ISSN: 00978930.

[31] N. Poh. “User-specific Score Normalization and Fusion for Biometric Person Recognition”. In: *Advanced Topics in Biometrics, World Scientific Publisher* (2011), pp. 401–418.

[32] N. Poh and S. Bengio. “How Do Correlation and Variance of Base-Experts Affect Fusion in Biometric Authentication Tasks?” In: *IEEE Transactions on Signal Processing* 53.11 (2005), pp. 4384–4396. ISSN: 1053-587X.

[33] L. Puente et al. “Biometrical Fusion – Input Statistical Distribution”. In: *Advanced Biometric Technologies*. Ed. by G. Chetty and J. Yang. InTech, 2011, pp. 87–110. ISBN: 978-953-307-487-0.

[34] D. Ruppert. “Modeling Univariate Distributions - Robust Estimation”. In: *Statistics and Data Analysis for Financial Engineering*. 978-1-4419-7786-1, 2010, pp. 117–119. ISBN: 978-1-4419-7786-1.

[35] J. C. Russ. “Image Enhancement in Spatial Domain”. In: *The Image Processing Handbook*. CRC Press, 2011, pp. 269–297. ISBN: 0-8493-7254-2.

[36] R. B. Rusu. “Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments”. PhD thesis. Aug. 2009, p. 260.

[37] Y. Su, S. Shan, and X. Chen. “Hierarchical Ensemble of Global and Local Classifiers for Face Recognition”. In: *Image Processing, IEEE* 18.8 (2009), pp. 1885–1896. ISSN: 1057-7149.

[38] X. Sun et al. “Fast and Effective Feature-Preserving Mesh Denoising”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.5 (2007), pp. 925–938. ISSN: 1077-2626.

[39] X. Tan and B. Triggs. “Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions”. In: *IEEE Transactions on Image Processing* 19.6 (2010), pp. 1635–1650. ISSN: 1941-0042.

[40] H. Tang et al. “3D Face Recognition Using Local Binary Patterns”. In: *Signal Processing* 93.8 (2013), pp. 2190–2198. ISSN: 0165-1684.

[41] M. A. Turk and A. P. Pentland. “Face recognition using eigenfaces”. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 591.1 (1991), pp. 586–591. ISSN: 1063-6919.

[42] J. Váňa et al. “Applying Fusion in Thermal Face Recognition”. In: *International Conference of the Biometrics Special Interest Group (BIOSIG)*. 2012. ISBN: 978-3-88579-290-1.

[43] C. Vielhauer, J. Dittmann, and S. Katzenbeisser. “Security and Privacy in Biometrics”. In: *Security and Privacy in Biometrics*. Ed. by P. Campisi. London: Springer London, 2013, pp. 25–43. ISBN: 978-1-4471-5229-3.

[44] P. Yang et al. “Face Recognition Using Ada-Boosted Gabor Features”. In: *Proceeding of 6th IEEE International Conference on Automatic Face and Gesture Recognition*. 2004, pp. 356–361. ISBN: 0-7695-2122-3.

[45] D. Zhang, J. Yang, and J. Y. Yang. “Is ICA Significantly Better than PCA for Face Recognition?” In: *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV’05)*. 2005, pp. 198–203. ISBN: 0-7695-2334-X.

europass

Europass Curriculum Vitae

Personal information

Surname(s) / First name(s)	**Štěpán Mráček**
Address(es)	Tábor 29, 616 00 Brno, Czech Republic
Email(s)	stepan.mracek@gmail.com
Date of birth	7. 10. 1985

Working experience

Period	2014 – present
Occupation	software engineer, analyst
Work	Development and improvement of biometric algorithms
Employer	TBS s.r.o. Brno, Czech Republic
Period	2006 – 2014
Occupation	software engineer, analyst
Work	Development of database applications in .NET environment
Employer	KUBITA s.r.o. Brno, Czech Republic

Education

Period	2010 – present
Study programme	Doctoral Degree Programme
School	Faculty of Information Technology Božetěchova 2 612 66 Brno, Czech Republic
Thesis theme	Face recognition
Period	2010
School	Gjøvik University College Teknologivn. 22 Gjøvik, Norway
Activity	Study within the frame of The ERASMUS Programme
Period	2008 – 2010
Academic degree	Master
Study programme	Information Technology, Intelligent Systems
School	Faculty of Information Technology Božetěchova 2 612 66 Brno, Czech Republic
Thesis theme	Biometric Recognition of 3D Faces
Period	2005 – 2008
Academic degree	Bachelor
Study programme	Information Technology

School

Faculty of Information Technology
Božetěchova 2
612 66 Brno, Czech Republic

Research interest

Biometric systems
Pattern recognition, machine learning, and classification
Computer graphics and image processing

University projects

2014 *Reliability and Security in IT*
VUT v Brně, FIT-S-14-2486
Research leader: Vojnar Tomáš

2013 *Innovation voucher - Touchless Biometric Systems s.r.o.*
TBS_CZ, 08072013-01
Research leader: Doležel Michal
Team leaders: Drahanský Martin, Dvořák Radim, Mráček Štěpán, Trhoň Adam, Váňa Jan

2012 *Intelligent assistance systems: multisensor processing and reliability analysis*
NATO, CBP.EAP.CLG 984, 2012
Research leader: Yanushkevich Svetlana
Team leaders: Drahanský Martin, Dvořák Radim, Mráček Štěpán, Váňa Jan

2012 *St. Anne´s University Hospital in Brno - International Clinical Research Center*; MŠMT, CZ.1.05/1.1.00/02.0123
Team leaders: Drahanský Martin

2010 *Tools and Methods for Video and Image Processing for the Fight against Terrorism*; MV, VG20102015006
Research leader: Zendulka Jaroslav
Team leaders: Drahanský Martin, Zemčík Pavel

2009 *Information Technology in Biomedical Engineering*
GAČR, GD102/09/H083
Research leader: Provazník Ivo
Team leaders: Drahanský Martin, Dušek Ladislav, Hanáček Petr, Holčík Jiří, Hřebíček Jiří, Kolářová Jana, Kozumplík Jiří, Zemčík Pavel

Skills

Mother tongue(s) **Czech**

Self-assessment
European level(*)

	Understanding		Speaking		Writing
	Listening	Reading	Spoken interaction	Spoken production	
English	B2 Independent user	C1 Proficient user	B2 Independent user	B2 Independent user	B1 Independent user

(*)*Common European Framework of Reference (CEF) level*

Professional skills: Knowledge of C, C++, C#, Java, Python, SQL, OpenCV, HTML+JS
Other skills: driving license

Brno, March 12, 2015
Štěpán Mráček